Google Research Blog

The latest news from Research at Google

Google Flu Trends gets a brand new engine

Posted: Friday, October 31, 2014

Posted by
Christian
Stefansen, Senior Software Engineer

Each year the flu kills thousands of people and affects millions around the world. So it's important that public health officials and health professionals learn about outbreaks as quickly as possible. In 2008 we launched Google Flu Trends in the U.S., using aggregate web searches to indicate when and where influenza was striking in real time. These models nicely complement other survey systems—they're more fine-grained geographically, and they're typically more immediate, up to 1-2 weeks ahead of traditional methods such as the CDC's official reports. They can also be incredibly helpful for countries that don't have official flu tracking. Since launching, we've expanded Flu Trends to cover 29 countries, and launched Dengue Trends in 10 countries.

The original model performed surprisingly well despite its simplicity. It was retrained just once per year, and typically used only the 50 to 300 queries that produced the best estimates for prior seasons. We then left it to perform through the new season and evaluated it at the end. It didn't use the official CDC data for estimation during the season—only in the initial training.

In the 2012/2013 season, we significantly overpredicted compared to the CDC's reported U.S. flu levels. We investigated and in the 2013/2014 season launched a retrained model (still using the original method). It performed within the historic range, but we wondered: could we do even better? Could we improve

the accuracy significantly with a more robust model that learns continuously from official flu data?

So for the 2014/2015 season, we're launching a new Flu Trends model in the U.S. that—like many of the best performing methods [1, 2, 3] in the literature—takes official CDC flu data into account as the flu season progresses. We'll publish the details in a technical paper soon. We look forward to seeing how the new model performs in 2014/2015 and whether this method could be extended to other countries.

As we've said since 2009, *"This system is not designed to be a replacement for traditional surveillance networks or supplant the need for laboratory-based diagnoses and surveillance."* But we do hope it can help alert health professionals to outbreaks early, and in areas without traditional monitoring, and give us all better odds against the flu.

Stay healthy this season!

Labels: Flu Trends, Health, Search
57 comments

Learning Statistics with Privacy, aided by the Flip of a Coin

Posted: Thursday, October 30, 2014

Posted by Úlfar Erlingsson,
Tech Lead Manager, Security Research

(Cross-posted on the Chromium Blog and the Google Online Security Blog)

At Google, we are constantly trying to improve the techniques we use to protect our users' security and privacy. One such project, RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response), provides a new state-of-the-art, privacy-preserving way to learn software statistics that we can use to better safeguard our users' security, find bugs, and improve the overall user experience.

Building on the concept of randomized response, RAPPOR enables learning statistics about the behavior of users' software while guaranteeing client privacy. The guarantees of differential

privacy, which are widely accepted as being the strongest form of privacy, have almost never been used in practice despite intense research in academia. RAPPOR introduces a practical method to achieve those guarantees.

To understand RAPPOR, consider the following example. Let's say you wanted to count how many of your online friends were dogs, while respecting the maxim that, on the Internet, nobody should know you're a dog. To do this, you could ask each friend to answer the question "Are you a dog?" in the following way. Each friend should flip a coin in secret, and answer the question truthfully if the coin came up heads; but, if the coin came up tails, that friend should always say "Yes" regardless. Then you could get a good estimate of the true count from the greater-than-half fraction of your friends that answered "Yes". However, you still wouldn't know which of your friends was a dog: each answer "Yes" would most likely be due to that friend's coin flip coming up tails.

RAPPOR builds on the above concept, allowing software to send reports that are effectively indistinguishable from the results of random coin flips and are free of any unique identifiers. However, by aggregating the reports we can learn the common statistics that are shared by many users. We're currently testing the use of RAPPOR in Chrome, to learn statistics about how unwanted software is hijacking users' settings.

We believe that RAPPOR has the potential to be applied for a number of different purposes, so we're making it freely available for all to use. We'll continue development of RAPPOR as a standalone open-source project so that anybody can inspect and test its reporting and analysis mechanisms, and help develop the technology. We've written up the technical details of RAPPOR in a report that will be published next week at the ACM Conference on Computer and Communications Security.

We're encouraged by the feedback we've received so far from academics and other stakeholders, and we're looking forward to additional comments from the community. We hope that everybody interested in preserving user privacy will review the technology and share their feedback at rappor-discuss@googlegroups.com

Labels: open source, publication, Security and Privacy
53 comments

HDR+: Low Light and High Dynamic

Range photography in the Google Camera App

Posted: Monday, October 27, 2014

Posted by Marc Levoy,
Google[x] Software Engineering Manager and Professor Emeritus, Stanford University

As anybody who has tried to use a smartphone to photograph a dimly lit scene knows, the resulting pictures are often blurry or full of random variations in brightness from pixel to pixel, known as image noise. Equally frustrating are smartphone photographs of scenes where there is a large range of brightness levels, such as a family photo backlit by a bright sky. In high dynamic range (HDR) situations like this, photographs will either come out with an overexposed sky (turning it white) or an underexposed family (turning them into silhouettes).

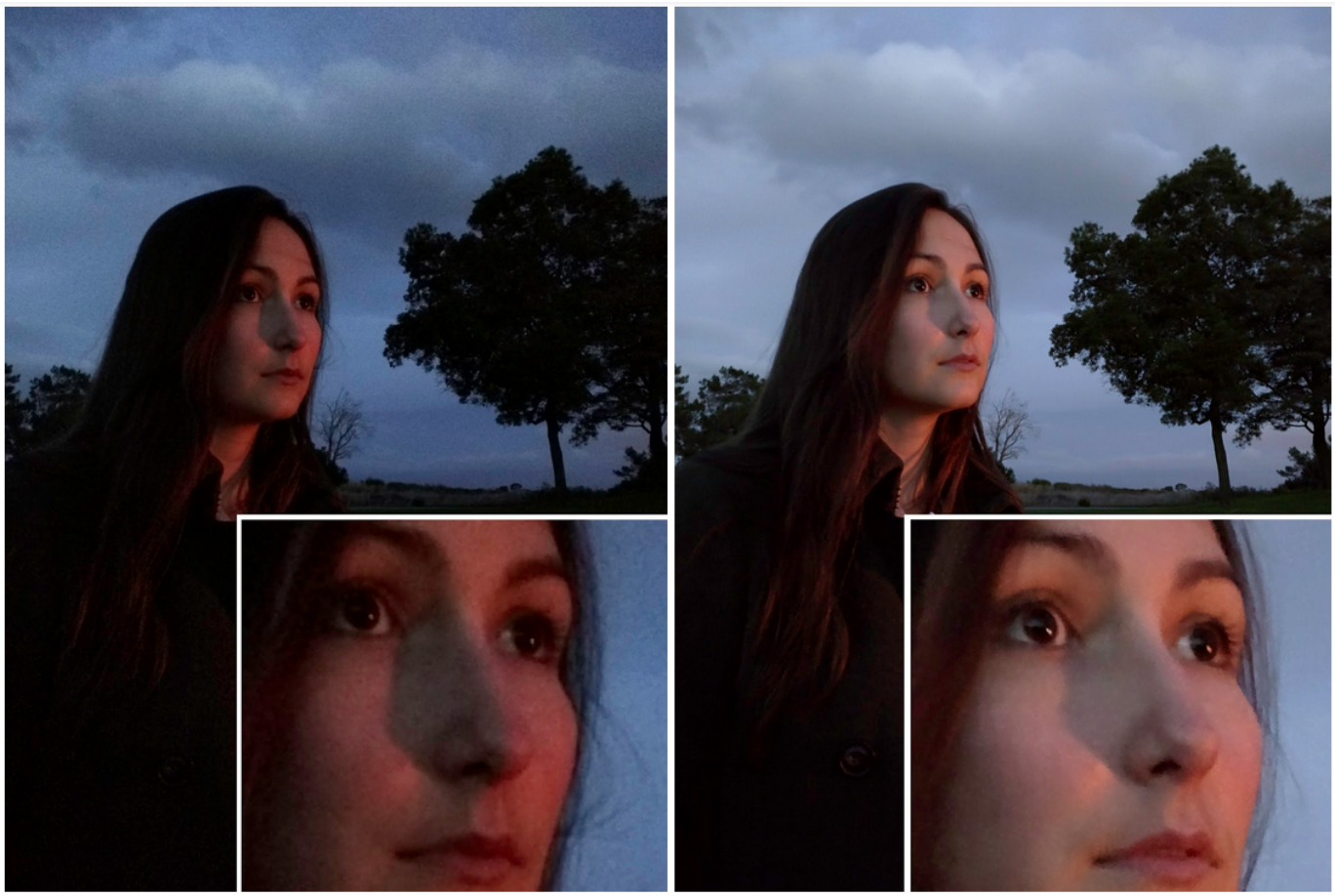
HDR+ is a feature in the Google Camera app for Nexus 5 and Nexus 6 that uses computational photography to help you take better pictures in these common situations. When you press the shutter button, HDR+ actually captures a rapid burst of pictures, then quickly combines them into one. This improves results in both low-light and high dynamic range situations. Below we delve into each case and describe how HDR+ works to produce a better picture.

Capturing low-light scenes

The camera on a smartphone has a small lens, meaning that it doesn't gather much light. If a scene is dimly lit, the resulting photograph will contain image noise. One solution is to lengthen the exposure time - how long the sensor chip collects light. This reduces noise, but since it's hard to hold a smartphone perfectly steady, long exposures have the unwanted side effect of blurring the shot. Devices with optical image stabilization (OIS) sense this "camera shake" and shift the lens rapidly to compensate. This allows longer exposures with less blur, but it can't help with really dark scenes.

HDR+ addresses this problem by taking a burst of shots with short exposure times, aligning them algorithmically, and replacing each pixel with the average color at that position across all the shots. Averaging multiple shots reduces noise, and using short exposures reduces blur. HDR+ also begins the

alignment process by choosing the sharpest single shot from the burst. Astronomers call this lucky imaging, a technique used to reduce the blurring of images caused by Earth's shimmering atmosphere.



A low light example is captured at dusk. The picture at left was taken with HDR+ off and the picture at right with HDR+ on. The HDR+ image is brighter, cleaner, and sharper, with much more detail seen in the subject's hair and eyelashes. Photos by Florian Kainz

Capturing high dynamic range scenes

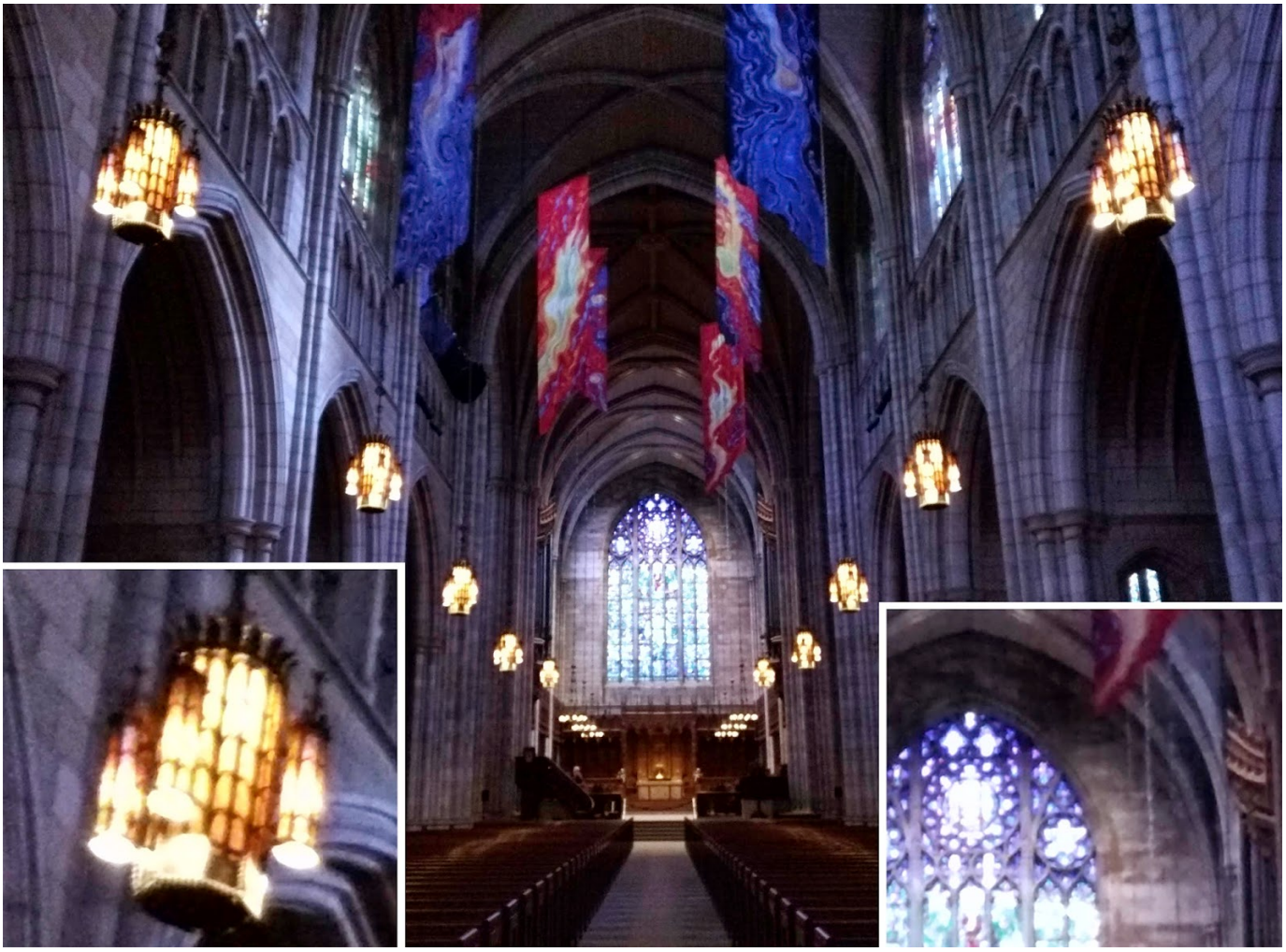
Another limitation of smartphone cameras is that their sensor chips have small pixels. This limits the camera's dynamic range, which refers to the span between the brightest highlight that doesn't blow out (turn white) and the darkest shadow that doesn't look black. One solution is to capture a sequence of pictures with different exposure times (sometimes called bracketing), then align and blend the images together. Unfortunately, bracketing causes parts of the long-exposure image to blow out and parts of the short-exposure image to be noisy. This makes alignment hard, leading to ghosts, double images, and other artifacts.

However, bracketing is not actually necessary; one can use the same exposure time in every shot. By using a short exposure HDR+ avoids blowing out highlights, and by combining enough shots it reduces noise in the shadows. This enables the

software to boost the brightness of shadows, saving both the subject and the sky, as shown in the example below. And since all the shots look similar, alignment is robust; you won't see ghosts or double images in HDR+ images, as one sometimes sees with other HDR software.

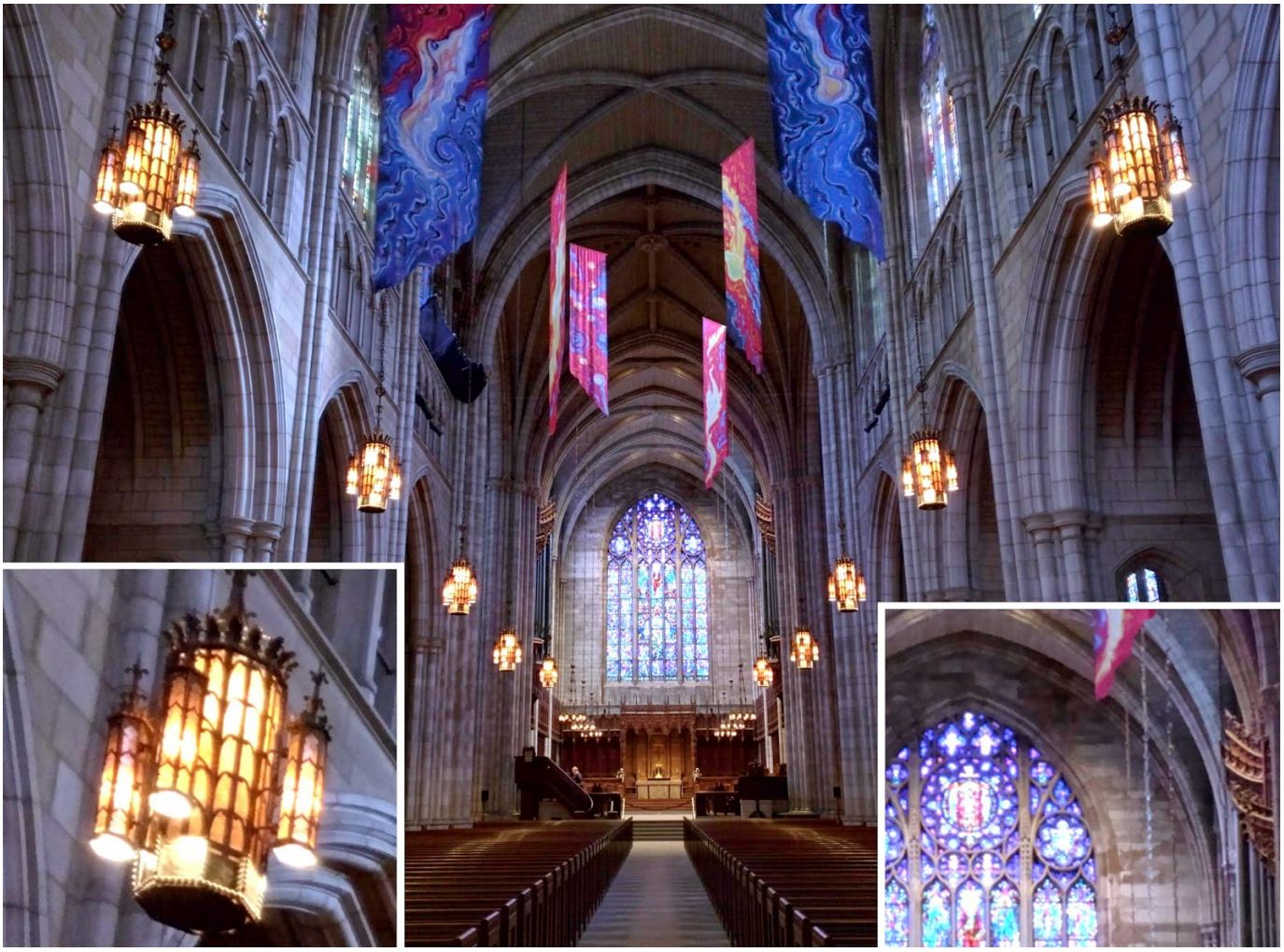
A classic high dynamic range situation. With HDR+ off (left), the camera exposes for the subjects' faces, causing the landscape and sky to blow out. With HDR+ on (right), the picture successfully captures the subjects, the landscape, and the sky. Photos by Ryan Geiss

Our last example illustrates all three of the problems we've talked about - high dynamic range, low light, and camera shake. With HDR+ off, a photo of Princeton University Chapel (shown below) taken with Nexus 6 chooses a relatively long 1/12 second exposure. Although optical image stabilization reduces camera shake, this is a long time to hold a camera still, so the image is slightly blurry. Since the scene was very dark, the walls are noisy despite the long exposure. Therefore, strong denoising is applied, causing smearing (below, left inset image). Finally, because the scene also has high dynamic range, the window at the end of the nave is blown out (below, right inset image), and the side arches are lost in darkness.



Click here to see the full resolution image. Photo by Marc Levoy

HDR+ mode performs better on all three problems, as seen in the image below: the chandelier at left is cleaner and sharper, the window is no longer blown out, there is more detail in the side arches, and since a burst of shots are captured and the software begins alignment by choosing the sharpest shot in the burst (lucky imaging), the resulting picture is sharp.

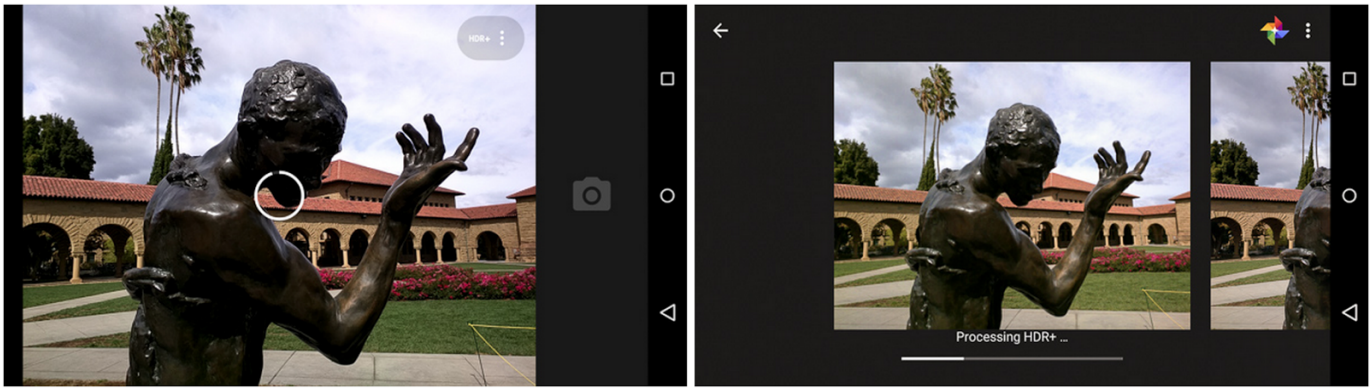


Click here to see the full resolution image. Photo by Marc Levoy

Here's an album containing these comparisons and others as high-resolution images. For each scene in the album there is a pair of images captured by Nexus 6; the first was was taken with HDR+ off, and the second with HDR+ on.

Tips on using HDR+

Capturing a burst in HDR+ mode takes between 1/3 second and 1 second, depending on how dark the scene is. During this time you'll see a circle animating on the screen (left image below). Try to hold still until it finishes. The combining step also takes time, so if you scroll to the camera roll right after taking the shot, you'll see a thumbnail image and a progress bar (right image below). When the bar reaches 100%, your HDR+ picture is ready.



Should you leave HDR+ mode on? We do. The only times we turn it off are for fast-moving sports, because HDR+ pictures take longer to capture than a single shot, or for scenes that are so dark we need the flash. But before you turn off HDR+ for these action shots or super-dark scenes, give it a try; we think you'll be surprised how well it works!

At this time HDR+ is available only on Nexus 5 and Nexus 6, as part of the Google Camera app.

Labels: Computational Photography, High Dynamic Range Imaging, Image Processing
181 comments

Helping teachers teach computer science

Posted: Friday, October 24, 2014

Posted by
Karen Parker,
Education Program Manager and Jason Ravitz, Education Evaluation Manager

(Cross-posted on the Google for Education Blog)

Since 2009, Google's CS4HS (Computer Science for High School) grant program has connected more than 12,000 computer science (CS) teachers with skills and resources to teach CS in fun and relevant ways. An estimated 600,000 students have been impacted by the teachers who have completed CS4HS professional development workshops so far.

Through annual grants, nearly 230 colleges and universities have hosted professional development workshops worldwide.

Grantees use the funds to develop CS curriculum and professional development workshops tailored for local middle and high school teachers. These workshops expose teachers to CS curriculum using real-world applications that spark students' curiosity. As feedback from those teachers rolls in, we want to share some highlights from what we've learned so far.

What went well:

- 89% of participants reported they would recommend their workshop to others
- 44% more participants reported a "high" or "very high knowledge" of CS after their workshop vs. before
- More than half of participants said they would use "most" or "all" of the activities or resources presented during their workshop.
- In 2014 the number of teachers who took part in a CS4HS professional development workshop increased by 50%, primarily due to the funding of multiple MOOCs.

Ways to make a bigger impact:

- Just 53% of participants said they felt a sense of community among the other workshop participants. Research by Joyce & Showers (2002) and Wiske, Stone, & Levinson (1993) shows that peer-to-peer professional development, along with ongoing support, helps teachers implement new content, retain skills, and create lasting change. We'll explore new ways to build community among participants as we plan future workshops.
- 83% of participants reported being Caucasian, which is consistent with the current demographics of CS educators. This indicates a need to increase efforts in diversifying the CS teacher population.
- Outcome measures show us that the most knowledge gains were among teachers who had no prior experience teaching CS or participating in CS professional development -- a population that made up just 30% of participants. While we see that the workshops are meeting a need, there remains an opportunity to develop materials geared toward more experienced CS teachers while also encouraging more new teachers to participate.

We know there are many challenges to overcome to improve the

state of CS teacher professional development. We look forward to sharing new ideas for working in partnership with the CS education community to help address those challenges, in particular by helping more teachers teach computer science.

At the University of Sydney CS4HS workshop teachers are learning how to teach Computer Science without a computer during a CS Unplugged activity.

Labels: Computer Science, MOOC, Professional Development

1 comment

Smart Autofill - Harnessing the Predictive Power of Machine Learning in Google Sheets

Posted: Monday, October 13, 2014

Posted by Konstantin Davydov, Software Engineer and Afshin Rostamizadeh, Research Scientist

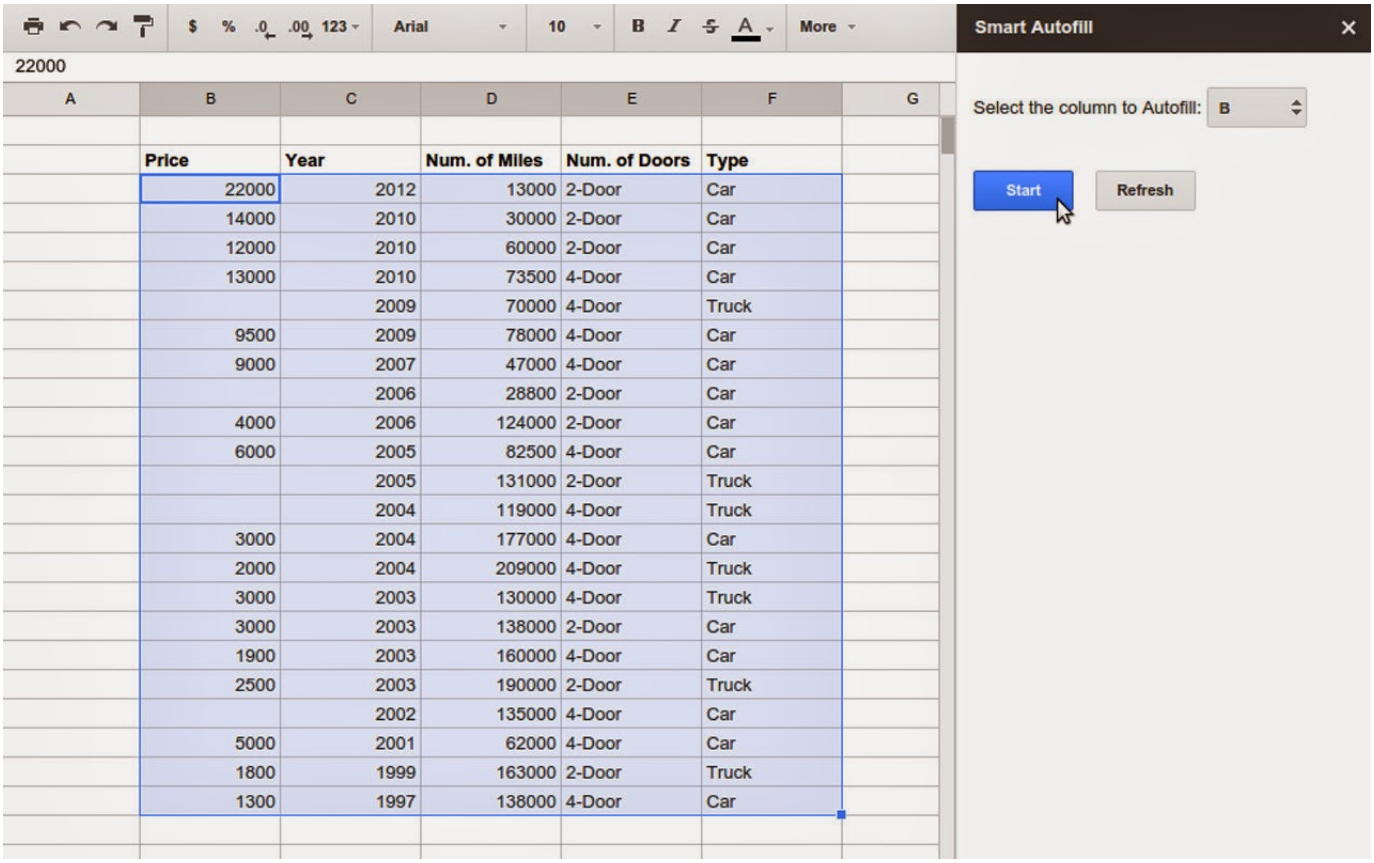
Much of Google's work on language, speech, translation, and visual processing relies on machine learning, where we construct and apply learning algorithms that make use of labeled data in order to make predictions for new data. What if you could leverage machine learning algorithms to learn patterns in your spreadsheet data, automatically build a model, and infer unknown values?

You can now use machine learning to make predictions in Google Sheets with the newly launched Smart Autofill Add-on. With a single click, Smart Autofill predicts the missing values of a partially filled column in your spreadsheet by using the data of other related columns. Smart Autofill uses the non-missing data to learn patterns and differs from the standard "Auto-fill" feature of Sheets, which attempts to fill in only simple patterns that it already knows (e.g. calendar dates, days of the week, ordered numbers).

As an example, in the screenshots below, we give four very simple characteristics of used vehicles (year, number of miles, number of doors, and type: car or truck) as well as the price for some of the vehicles. Since the prices are probably correlated with the characteristics of the vehicle, we can use Smart Autofill to estimate what the missing prices should be. The rows that do contain a price will be used as examples to learn from in order to fill in the rows with a missing price.

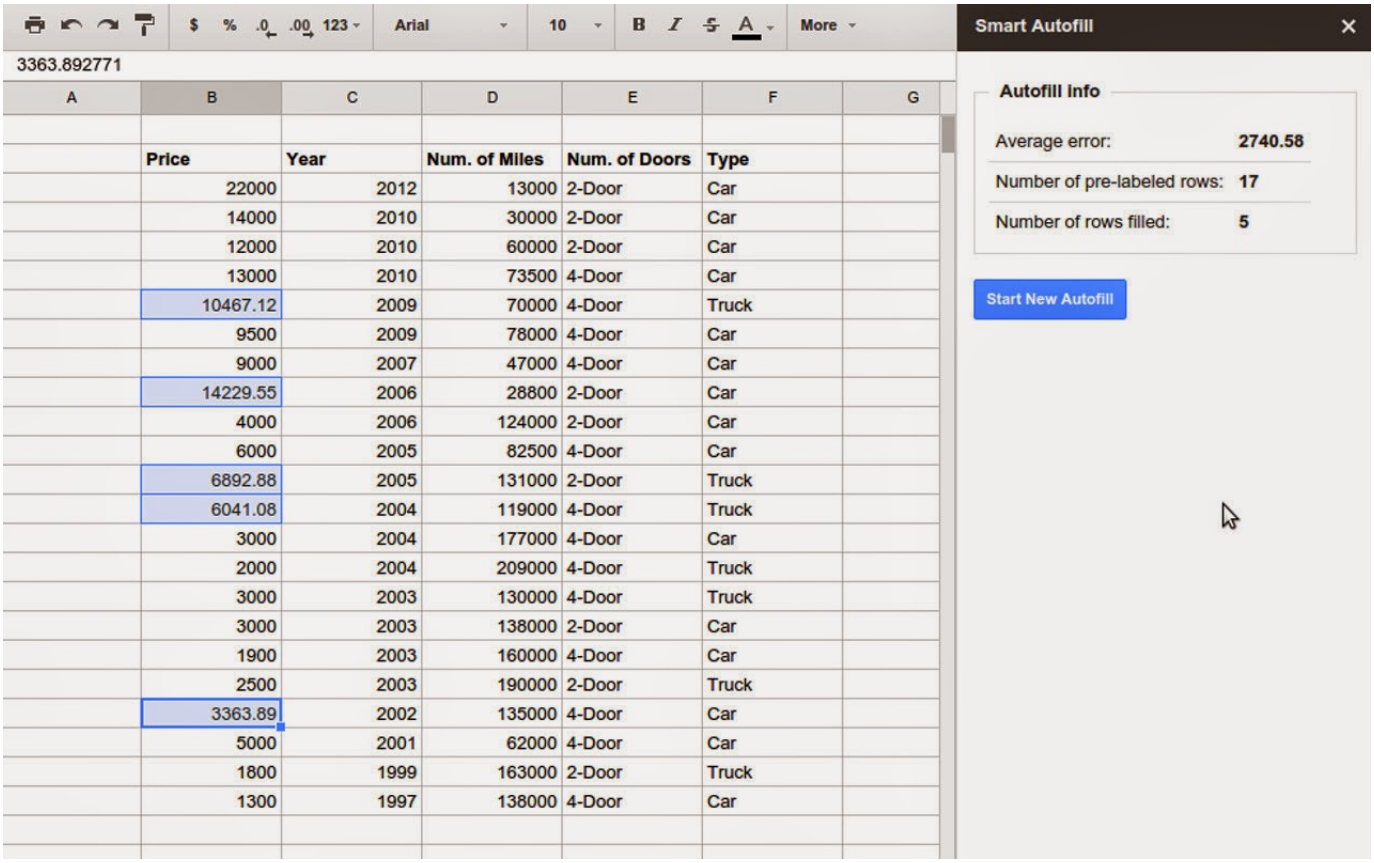
Smart Autofill uses Google's cloud-based machine learning service Prediction API, which trains several linear as well as non-linear classification and regression models. The best model is automatically chosen for your problem by finding the one with the smallest misclassification error (for categorical data) or root-mean-squared error (for numeric data) calculated by using cross-validation on the labeled (non-empty) set of examples.

To use Smart Autofill, after following the installation procedure, simply select "Add-ons > Smart Autofill > Start" which will open a sidebar. Select a block of data that includes the column to Autofill and click "Next". Finally, from the selected data, choose a target column to Autofill and click "Start" (Figure 1). Now just sit back as Smart Autofill does its work and fills in the missing values (Figure 2).



	Price	Year	Num. of Miles	Num. of Doors	Type
	22000	2012	13000	2-Door	Car
	14000	2010	30000	2-Door	Car
	12000	2010	60000	2-Door	Car
	13000	2010	73500	4-Door	Car
		2009	70000	4-Door	Truck
	9500	2009	78000	4-Door	Car
	9000	2007	47000	4-Door	Car
		2006	28800	2-Door	Car
	4000	2006	124000	2-Door	Car
	6000	2005	82500	4-Door	Car
		2005	131000	2-Door	Truck
		2004	119000	4-Door	Truck
	3000	2004	177000	4-Door	Car
	2000	2004	209000	4-Door	Truck
	3000	2003	130000	4-Door	Truck
	3000	2003	138000	2-Door	Car
	1900	2003	160000	4-Door	Car
	2500	2003	190000	2-Door	Truck
		2002	135000	4-Door	Car
	5000	2001	62000	4-Door	Car
	1800	1999	163000	2-Door	Truck
	1300	1997	138000	4-Door	Car

Figure 1: Highlighting the dataset and selecting the target column.



	Price	Year	Num. of Miles	Num. of Doors	Type
	22000	2012	13000	2-Door	Car
	14000	2010	30000	2-Door	Car
	12000	2010	60000	2-Door	Car
	13000	2010	73500	4-Door	Car
	10467.12	2009	70000	4-Door	Truck
	9500	2009	78000	4-Door	Car
	9000	2007	47000	4-Door	Car
	14229.55	2006	28800	2-Door	Car
	4000	2006	124000	2-Door	Car
	6000	2005	82500	4-Door	Car
	6892.88	2005	131000	2-Door	Truck
	6041.08	2004	119000	4-Door	Truck
	3000	2004	177000	4-Door	Car
	2000	2004	209000	4-Door	Truck
	3000	2003	130000	4-Door	Truck
	3000	2003	138000	2-Door	Car
	1900	2003	160000	4-Door	Car
	2500	2003	190000	2-Door	Truck
	3363.89	2002	135000	4-Door	Car
	5000	2001	62000	4-Door	Car
	1800	1999	163000	2-Door	Truck
	1300	1997	138000	4-Door	Car

Figure 2: After clicking "Start" a model is trained and applied to automatically fill in the missing values of the target column. Note, the estimated error of the model is reported in the sidebar.

An estimate of the error-rate of the model (based on the non-missing data) is shown in the sidebar after the missing values are filled. The accuracy of Smart Autofill (as well as the accuracy of the estimated error) depends on many factors, including the amount and quality of the data provided. While not all datasets will be ideally suited for machine learning, we hope our more in-depth tutorial will provide an idea of the range of problems where Smart Autofill can be effective.

While the vehicle pricing example is relatively simple (in reality used vehicle prices are a function of more than just four variables), more complex datasets could have many more non-target columns as well as data rows. Also, the target column does not need to be numeric, since Smart Autofill can also predict categorical values (i.e. in the car example the target column value could have contained the categories "expensive", "moderate", "affordable" instead of price). Other illustrative scenarios include:

- You have a spreadsheet that holds the results of a customer survey, but one of the columns (e.g. "overall satisfaction 1-5") has some missing values. If the other columns of the survey can help indicate

overall satisfaction then you can try using Smart Autofill to estimate the missing values.

- You keep a spreadsheet of restaurants that you've visited and their characteristics (type: Italian, ambiance: quiet, cost: $$$, etc.) and whether you enjoyed the restaurant or not. Now you can add the characteristics of new restaurants to your spreadsheet and use Smart Autofill to guess at which ones you might enjoy.

The example dataset and more detailed tutorial for the add-on can be found here. We hope you discover new and useful ways to incorporate the predictive power of machine learning with your data.

Labels: Google Sheets, Machine Learning
103 comments

All the News that's Fit to Read: A Study of Social Annotations for News Reading

Posted: Wednesday, October 08, 2014

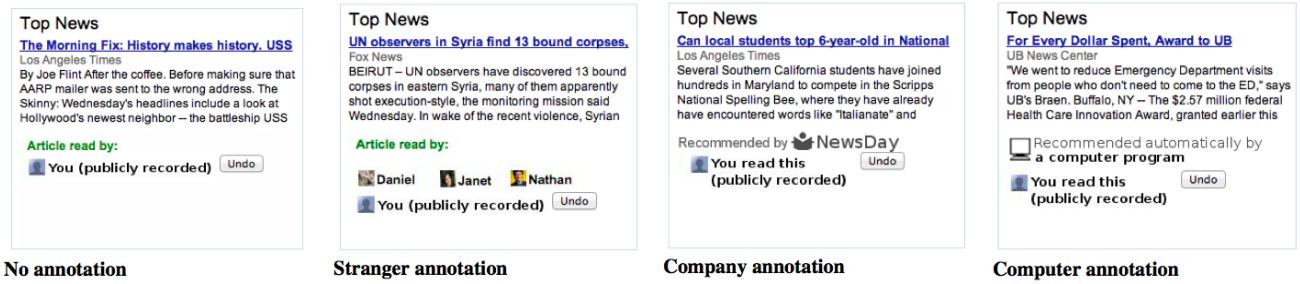
Posted by Chinmay Kulkarni, Stanford University Ph.D candidate and former Google Intern, and Ed H. Chi, Google Research Scientist

News is one of the most important parts of our collective information diet, and like any other activity on the Web, online news reading is fast becoming a social experience. Internet users today see recommendations for news from a variety of sources; newspaper websites allow readers to recommend news articles to each other, restaurant review sites present other diners' recommendations, and now several social networks have integrated social news readers.

With news article recommendations and endorsements coming from a combination of computers and algorithms, companies that publish and aggregate content, friends and even complete strangers, how do these explanations (i.e. why the articles are shown to you, which we call "annotations") affect users' selections of what to read? Given the ubiquity of online social

annotations in news dissemination, it is surprising how little is known about how users respond to these annotations, and how to offer them to users productively.

In All the News that's Fit to Read: A Study of Social Annotations for News Reading, presented at the 2013 ACM SIGCHI Conference on Human Factors in Computing Systems and highlighted in the list of influential Google papers from 2013, we reported on results from two experiments with voluntary participants that suggest that social annotations, which have so far been considered as a generic simple method to increase user engagement, are not simple at all; social annotations vary significantly in their degree of persuasiveness, and their ability to change user engagement.



News articles in different annotation conditions

The first experiment looked at how people use annotations when the content they see is not personalized, and the annotations are not from people in their social network, as is the case when a user is not signed into a particular social network. Participants who signed up for the study were suggested the same set of news articles via annotations from strangers, a computer agent, and a fictional branded company. Additionally, they were told whether or not other participants in the experiment would see their name displayed next to articles they read (i.e. "Recorded" or "Not Recorded").

Surprisingly, annotations by unknown companies and computers were significantly more persuasive than those by strangers in this "signed-out" context. This result implies the potential power of suggestion offered by annotations, even when they're conferred by brands or recommendation algorithms previously unknown to the users, and that annotations by computers and companies may be valuable in a signed-out context. Furthermore, the experiment showed that with "recording" on, the overall number of articles clicked decreased compared to participants *without* "recording," regardless of the type of annotation, suggesting that subjects were cognizant of how they appear to other users in social reading apps.

If annotations by strangers is not as persuasive as those by computers or brands, as the first experiment showed, what about the effects of *friend* annotations? The second experiment examined the signed-in experience (with Googlers as subjects) and how they reacted to social annotations from friends, investigating whether personalized endorsements help people discover and select what might be more interesting content.

Perhaps not entirely surprising, results showed that friend annotations are persuasive and improve user satisfaction of news article selections. What's interesting is that, in post-experiment interviews, we found that annotations influenced whether participants read articles primarily in three cases: first, when the annotator was above a threshold of social closeness; second, when the annotator had subject expertise related to the news article; and third, when the annotation provided additional context to the recommended article. This suggests that social context and personalized annotation work together to improve user experience overall.

Some questions for future research include whether or not highlighting expertise in annotations help, if the threshold for social proximity can be algorithmically determined, and if aggregating annotations (e.g. "110 people liked this") help increases engagement. We look forward to further research that enable social recommenders to offer appropriate explanations for why users should pay attention, and reveal more nuances based on the presentation of annotations.

Labels: conferences, HCI, Publications
20 comments

Announcing the Google CS Engagement Small Awards Program

Posted: Monday, October 06, 2014

Posted by
Leslie Yeh
Johnson, University Relations

(cross-posted on the Google for Education blog)

College students are more interested than ever in studying computer science. There has been an unprecedented increase in enrollment in Computer Science undergraduate programs over

the past six years. Harvard University's popular introductory CS course CS50 has recently claimed the spot as the most enrolled course on campus. An astounding 50% of Harvey Mudd's graduates received engineering degrees this year. However, while the overall number of students in introductory computer science courses continue to climb, the number of students who go on to complete undergraduate degrees in this field, particularly among women and under-represented minorities, does not match this increase in individual course enrollment (2013 Taulbee Survey).

Recent findings show that while students may begin a CS degree program, retaining students after their first year remains an issue. Research indicates that one of the strongest factors in the retention of students in undergraduate CS degrees is early exposure to engaging courses and course material, such as high quality assignments that are meaningful and relevant to the student's life or classroom activities that encourage student-to-student interaction. When an instructor or department imbeds these practices into the introductory CS classroom, students remain excited about CS and are more likely to complete their undergraduate CS degree.

At Google we believe in the importance of preparing the next generation of computer scientists. To this end, we've created the CS Engagement Small Grants Program to support educators teaching introductory computer science courses in reaching their engagement and retention goals. We'll give unrestricted gifts of $5,000 to the selected applicants' universities, towards the execution of engaging CS1 or CS2 courses in the 2014-2015 school year. We encourage educators who are teaching CS1 and CS2 courses at the post-secondary level to apply to the Google CS Engagement Small Grants Program. Applications will be accepted through **November 15, 2014** and will be evaluated on an ongoing basis. If you're interested in applying, please check out the Call for Proposals.

Labels: Awards, Computer Science, Education

11 comments

Newer Posts Home Older Posts

Official Google Blog
Android Blog
Developers Blog
Public Policy Blog
Chrome Blog
Ads Developer Blog
Student Blog
Lat Long Blog
Android Developers Blog